

UNITED STATES PATENT APPLICATION

DELAY LOCKED LOOP "ACTIVE COMMAND" REACTOR

INVENTOR
Debra M. Bell

Schwegman, Lundberg, Woessner, & Kluth, P.A.
1600 TCF Tower
121 South Eighth Street
Minneapolis, Minnesota 55402
ATTORNEY DOCKET 303.752US1
MICRON 00-1197

09903227.071101
T0FF20" 222E0660

DELAY LOCKED LOOP "ACTIVE COMMAND" REACTOR

Technical Field of the Invention

The present invention relates to signal generation in integrated circuits, and in particular to signal generation using a delay lock loop.

Background of the Invention

Delay locked loops (DLLs) are often used in integrated circuits (ICs) to generate an internal clock signal. In a typical DLL, the internal clock signal is generated by applying a delay to a system clock or an external clock signal. The DLL automatically adjusts the delay to keep the internal and external clock signals synchronized.

In some integrated circuit devices, such as dynamic random access memory (DRAM) devices, a DLL is normally used to provide a timing signal for certain operations of the memory device. For example, in some memory devices, the internal clock signal generated by the DLL can be used as capture clock signal, or a strobe signal during a READ or a WRITE mode.

A traditional memory device has a number of memory cells to store data. To retrieve the stored data, a READ mode is performed. Typically, the READ mode includes two steps. First, the memory device activates a so-called ACTIVE command signal during an ACTIVE mode to "open" or activate the memory cells. Next, a READ command signal is activated to access the memory cells to read the stored data. In the traditional memory device, activating the memory cells during the ACTIVE mode demands a high amount of current. This causes the supply voltage of the memory device to drop. The drop in the supply voltage changes the voltage supplied to the DLL of the memory device. The change in the voltage supply of the DLL causes the external and internal clock signals to be out of synchronism.

The DLL attempts to keep the internal and external clock signals synchronized by adjusting the delay to compensate for any variation in operating conditions such as the drop in the supply voltage. Reducing the effect of the change in operating conditions of the DLL such as the voltage drop during the ACTIVE mode is desirable.

For the reasons stated above, and for other reasons stated below which will become apparent to those skilled in the art upon reading and understanding the present specification, there is a need in the art for an improved DLL.

Summary of the Invention

5 The present invention includes a delay locked loop (DLL) that applies an amount of delay to an external clock signal to generate a first and a second delayed signal. One of the delayed signals is selected as an internal clock signal. The first and second delayed signals have different delays in relation to the external clock signal. If a
10 change in operating condition of the DLL occurs, such as a change in the supply voltage during an operational mode of the memory device such as an ACTIVE, a READ or a REFRESH mode, the DLL immediately selects another delayed signal as a new internal clock signal to compensate for the change in operating condition before a phase detector of the DLL detects the change.

15 In one aspect, the DLL includes a delay line that has an input for receiving an external clock signal and multiple outputs for providing multiple delayed signals. A selector of the DLL selects one of the multiple delayed signals as an internal clock signal. The multiple delayed signals have different delays in relation to the external clock signal. The DLL also includes a command react circuit connected to the selector.
20 The command react circuit is capable of activating a command set signal for enabling the selector to select a different delayed signal among the multiple delayed signals based on a command signal.

In another aspect, a method of operating a DLL is provided. The method includes generating multiple delayed signals by delaying an external clock signal. The
25 first delayed signal among the multiple delayed signals is selected to be an internal clock signal, which is synchronized with the external clock signal. When a command signal is activated, a second delayed signal among the multiple delayed signals is selected to be the internal clock signal. The selection of second delay signal occurs before the external and internal are being detected as out of synchronism. After the
30 command signal is deactivated, the first signal is again selected as the internal clock

signal. The external and internal signals are synchronized in subsequent DLL operations.

Brief Description of the Drawings

- 5 Figure 1 is a block diagram of a DLL according to one embodiment the invention.
- Figure 2 is timing diagram showing timing relationships of various signals of the DLL of Figure 1.
- Figure 3 is a block diagram of a delay line and a controller according to one embodiment of the invention.
- 10 Figures 4A-E are block diagrams of delay lines connected to selectors according to various embodiments of the invention.
- Figure 5 is a schematic diagram of a command react circuit according to one embodiment of the invention.
- 15 Figure 6 is a block diagram of a memory device according to one embodiment of the invention.
- Figure 7 is a block diagram showing in more detail a portion of the memory device of Figure 6.
- Figure 8 is a block diagram of a system according to one embodiment of the invention.
- 20

Detailed Description of the Invention

The following detailed description refers to the accompanying drawings which form a part hereof, and which is shown, by way of illustration specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical and electrical changes may be made without departing from the spirit and scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the invention is defined only by the appended claims.

25

30

Figure 1 is a block diagram of a DLL 100 according to one embodiment of the invention. In one embodiment, DLL 100 is a digital DLL. DLL 100 includes an input at node 101 and an output at node 199. Input 101 receives an external clock signal XCLK and output 199 provides an internal clock signal DLLCLK.

5 DLL 100 also includes a delay line 112. Delay line 112 has an input connected to node 101 to receive the XCLK signal and two outputs 114 and 116. Output 114 provides a first delayed signal DLLCK1. Output 116 provides a second delayed signal DLLCK0. Delay line 112 comprises a plurality of delay stages 113-0 to 113-N connected in series, where N is the total number of the delay stages. Each of the delay
10 stages can delay a signal by a predetermined delay time.

Each of the delay stages 113-0 to 113-N provides a delay (delay time). Thus the amount of delay applied to the XCLK signal is equal to the number of delayed stages being used by delay line 112 multiplied by the delay of each delay stage. In the embodiment represented by Figure 1, the DLLCK1 signal is provided by the last delay
15 stage 113-N and the DLLCK0 signal is provided by the next to last delay stage 113 (N-1). Thus, the difference in the number of delay stages used to produce the DLLCK0 and DLLCK1 signals is equal to one delay stage. Therefore, the difference in delay between the first and second delayed signals DLLCK0 and DLLCK1 is equal to the delay of one delay stage.

20 Delay line 112 connects to a controller 124 through a plurality of control taps 119-0 to 119-N. Controller 124 receives a shift left signal SL on line 142, a shift right signal SR on line 143, and a phase lock signal PHLOCK on line 144. The SL and SR are also referred to as shifting signals. Based on the shifting signals SL and SR, controller 124 adjusts an amount of delay applied to the XCLK signal by delay line 112
25 to provide the first and second delayed signals DLLCK0 and DLLCK1. Controller 124 adjusts the delay by causing delay line 112 to either increase or decrease the amount of delay to the XCLK signal. By adjusting the delay, the phase relationships between the XCLK signal and the DLLCK0 and DLLCK1 signals at outputs 114 and 116 are also adjusted accordingly.

The DLLCK0 and DLLCK1 signals at outputs 114 and 116 of delay line 112 are provided as inputs to a selector or multiplexor (MUX) 130. MUX 130 has inputs A and B. Input A connects to output 116 to receive the DLLCK0. Input B connects to output 114 to receive the DLLCK1 signal. An output of MUX 130 connects to node 199 to provide the DLLCLK signal. The DLLCLK signal at output node 199 is selected based on a COMMAND_SET signal at line 122. Based on the COMMAND_SET signal, MUX 130 selects between the DLLCK0 and DLLCK1 signals and passes the selected signal to node 199 as the DLLCLK signal. The DLLCLK signal is fed back to a phase detector 150.

Phase detector 150 includes a first input connected to node 101 to receive the XCLK signal. A second input of phase detector 150 connects to a feedback path or line 155 to receive the DLLCLK signal. Phase detector 150 also includes a plurality of outputs connected to lines 142, 143, 144 and 145. Lines 142, 143 and 144 provide the SL, SR, and PHLOCK signals. Line 145 provides a phase detect signal PHDETECT.

When phase detector 150 detects a difference in phase between the XCLK and DLLCLK signals, i.e., when the XCLK and DLLCLK signals are not synchronized, phase detector 150 activates the SL or SR signal. In the embodiment represented by Figure 1, the PHDETECT signal is activated when the XCLK and DLLCLK signals are out of synchronism and the XCLK signal lags the DLLCLK signal. In some other embodiments, the PHDETECT signal is activated when the XCLK and DLLCLK signals are out of synchronism and the XCLK signal leads the DLLCLK signal.

When the XCLK and DLLCLK signals are synchronized, i.e., when the XCLK and DLLCLK signals have the same phase, phase detector 150 activates the PHLOCK signal and deactivates the SL, SR and PHDETECT signals. In summary, the PHLOCK signal is activated when the XCLK and the DLLCLK signals are synchronized. The PHDETECT signal is activated when the XCLK and the DLLCLK signals are not synchronized.

The PHLOCK and PHDETECT signals are provided to a command react circuit 140. Command react circuit 140 has a first and a second input connected to lines 144 and 145 to receive the PHLOCK and PHDETECT signals. A third input of command

react circuit 140 connects to line 146 for receiving a command signal COMMAND. Command react circuit 140 operates in a fashion such that when the COMMAND signal is at a high signal level (HIGH), the COMMAND_SET signal is forced HIGH regardless of the state of the PHLOCK and PHDETECT signals. When the
5 COMMAND signal is at a low signal level (LOW) and either the PHLOCK or PHDETECT signal is HIGH, the COMMAND_SET signal is forced LOW. In a normal DLL operation, the COMMAND_SET signal is LOW.

In general, DLL 100 applies a delay to the XCLK signal to generate the DLLCLK signal. The DLLCLK signal is constantly compared with the XCLK signal
10 by phase detector 150. When the rising edges of the XCLK and DLLCLK signals are not lined up, i.e., when the XCLK and DLLCLK signals are not synchronized, DLL 100 adjusts the delay applied to the XCLK signal accordingly to line up the edges. When the edges are lined up, i.e., when the XCLK and DLLCLK signals are synchronized, DLL 100 stops the adjusting and locks the DLL. The DLL normally remains in the lock
15 position.

The detailed operation of DLL 100 is better understood with the description of a timing diagram of Figure 2. Figure 2 is timing diagram showing the timing relationships of the XCLK, DLLCLK, COMMAND, COMMAND_SET, PHDETECT, and PHLOCK signals of DLL 100. In Figure 2, it is shown that DLL 100 is locked
20 from time T0 to time T1. In other words, between times T0 and T1, the XCLK and DLLCLK signals are synchronized and the DLLCLK signal is selected from the DLLCK1 signal as indicated in Figure 2 by DLLCLK=DLLCK1. Between times T0 and T1, because the XCLK and DLLCLK signals are synchronized, the PHDETECT signal is not activated (LOW) and the signal and the PHLOCK signal is activated
25 (HIGH). The COMMAND signal is not activated (LOW) between times T0 and T1.

At time T1, the COMMAND signal is activated, i.e., it switches from LOW to HIGH. This causes the COMMAND_SET signal to be HIGH, which causes MUX 130 to deselect the DLLCK1 signal and to select the DLLCK0 signal. The DLLCK0 signal is passed to node 199 as the new DLLCLK signal as indicated in Figure 2 by
30 DLLCLK=DLLCK0. Because the DLLCK0 is not the same as the DLLCK1 signal

(having different delay), switching the DLLCLK signal from the DLLCK1 signal to the DLLCK0 signal causes the DLLCLK and the XCLK to be out of synchronism or out of phase. In Figure 2, this out of synchronism is indicated by D1, which is the difference in phase between the edges of the XCLK and the DLLCLK signals.

5 In the embodiment represented by Figure 2, rising edge 202 of the DLLCLK signal comes before time T1 because the amount of delay introduced to the DLL by the activation of the COMMAND signal is less than the amount of delay that is taken from the DLL by the switching from the DLLCK1 signal to the DLLCK0 signal. However, in other embodiments, rising edge 202 of the DLLCLK signal comes after time T1 if the
10 amount of delay introduced to the DLL by the activation of the COMMAND signal is more than the amount of delay that is taken from the DLL by the switching from the DLLCK1 to the DLLCK0 signal.

As shown in Figure 2, at time T1, the XCLK and the DLLCLK signals become out of synchronism because of the activation of the COMMAND signal at time T1. At
15 time T1, phase detector 150 has not detect the difference in phase between the XCLK and DLLCLK signals. Hence the PHDETECT signal is still LOW at time T1. At a certain time after time T1, for example at time T2, the phase difference between the XCLK and the DLLCLK signals is detected by phase detector 150. At this time (T2), the XCLK signal lags the DLLCLK signal because the amount of delay introduced to
20 the DLL by the activation of the COMMAND signal is less than the amount of delay that is taken from the DLL by the switching from the DLLCK1 signal to the DLLCK0 signal. Since the XCLK signal lags the DLLCLK signal, phase detector 150 deactivates the PHLOCK signal and activates the SR or SL signal and the PHDETECT signal to begin the adjusting process in an attempt to put the XCLK and DLLCLK back to
25 synchronism. The adjustment process is performed between times T2 and T4 as indicated in Figure 2 as ADJUST.

After the COMMAND signal is activated at time T1, it is deactivated at a certain time after time T1. For example, at time T3, the COMMAND signal is deactivated.

The deactivation of the COMMAND signal at time T3 allows the COMMAND_SET
30 signal to be forced LOW when either the PHDETECT or the PHLOCK signal is

asserted (HIGH). When the COMMAND signal is deactivated at time T3, the
COMMAND_SET signal is forced LOW because the PHDETECT signal has been
asserted or activated HIGH since time T2. When the COMMAND_SET signal is forced
LOW at time T3, it causes MUX 130 to deselect the DLLCK0 signal and reselect the
5 DLLCK1 signal as the DLLCLK signal at time T3 as indicated in Figure 2 by
DLLCLK=DLLCK1.

In the embodiment represented by Figure 2, between times T2 and T3, the
PHDETECT signal is HIGH and the PHLOCK signal is LOW because the XCLK and
DLLCLK (DLLCK0) signals are not synchronized. Specifically, between times T2 and
10 T3, the XCLK signal lags the DLLCLK signal. However, in some other embodiments,
between times T2 and T3, the XCLK and DLLCLK (DLLCK0) signals may be
synchronized. If this occurs, at time T2, the PHDETECT signal will not be activated
(will stay LOW), however, the PHLOCK signal will be activated HIGH between times
T2 and T3. Thus, in the other embodiments, when the COMMAND is deactivated at
15 time T3 and the PHLOCK signal is HIGH, the COMMAND_SET signal is still forced
LOW allowing MUX 130 to reselect the DLLCK1 signal as the DLLCLK signal.

When the DLLCK1 signal is re-selected at time T3, the situation is once again
similar to the situation when the DLLCLK signal was switched from the DLLCK1
signal to DLLCK0 signal as described above. Switching from the DLLCK0 signal back
20 to the DLLCK1 signal at time T3 varies the delay applied to the XCLK signal. Hence,
the signal relationship between the XCLK and the DLLCLK changes. The change is
indicated by D2 at time T3. Rising edge 204 of the DLLCLK signal D2 comes after
time T3 because switching from the DLLCK0 signal to DLLCK1 signal adds more
delay to the DLLCLK signal.

25 Between times T2 and T3, the adjustment process continues. At a certain time,
for example at time T4, the XCLK and DLLCLK signals are synchronized. At this
time, the DLLCLK signal is still selected from the DLLCK1 signal as indicated in
Figure 2 by DLLCLK=DLLCK1. In response to this synchronism, phase detector 150
deactivates the PHDETECT signal and activates the PHLOCK signal to lock the DLL.

In the embodiment represented by Figure 2, the PHDETECT and PHLOCK signals make transition between LOW and HIGH one clock cycle after an event when either the XCLK and DLLCLK signals become synchronized or become out of synchronism. However, in some other embodiments, the PHDETECT and PHLOCK signals can make transition at certain time that is one or more clock cycles after the XCLK and DLLCLK signals become synchronized or become out of synchronism. The PHDETECT and PHLOCK signals can also make a transition between LOW to HIGH at the same time the XCLK and DLLCLK signal are synchronized or become out of synchronism. Further, the adjusting process in Figure 2 lasts one or more clock cycles.

10 The timing relationships of elements of DLL 100 when it operates has been described with the timing diagram represented by Figure 2. The following description describes the operation of DLL 100 with emphasis on functions of elements of DLL 100 as shown in Figure 1. In operation, referring to Figure 1, DLL 100 receives the XCLK signal at node 101. Delay line 112 applies an initial amount of delay to the XCLK signal by using an initial or a predetermined number of delay stages 113 0-N. The predetermined number of delay stages is initially set by controller 124. In subsequent actions of the operation, controller 124 selects a different number of delay stages to adjust, i.e., to increase or decrease the amount of delay by increasing or decreasing the number of delay stages. Thus, the amount of delay is proportional to the number of the selected delay stages through which the XCLK signal propagates. In other words, when the number of delay stages increases, the amount of delay applied to the XCLK signal is increased. Conversely, when the number of delay stages decreases, the amount of delay is decreased. After propagating through the number delay stages, the XCLK signal becomes the DLLCK0 and DLLCK1 signals at outputs 114 and 116, thus, the DLLCK0 and DLLCK1 are delayed versions of the XCLK signal.

In the embodiment represented by Figure 1, the COMMAND_SET signal is normally LOW. MUX 130 operates in a fashion such that when the COMMAND_SET signal is LOW, it selects the signal at input B and passes it to node 199. In this case, MUX 130 selects the DLLCK1 signal and pass it to node 199 as the DLLCLK signal. The DLLCLK signal is fed back to phase detector 150. Phase detector 150 compares

the relative edges of the DLLCLK and XCLK signals. If the XCLK and DLLCLK signals are not synchronized, phase detector 150 activates either the SL or SR signal based on the phase difference between the XCLK and DLLCLK signals.

Controller 124 receives either the SL or SR signal and adjusts the amount of delay applied to the XCLK signal. In adjusting the amount of delay, controller 124 causes delay line 112 to increase or decrease the amount of delay. To increase the amount of delay, delay line 112 adds one or more delay stages to the number of delay stages being used. To decrease the amount of delay, delay line 112 removes one or more delay stages from the number of delay stages being used. The increase or decrease in the amount of delay changes the DLLCLK signal accordingly. After the DLLCLK signal is changed, phase detector 150 compares the DLLCLK and XCLK signals again. The phase comparison and delay adjustment process repeats until the DLLCLK and XCLK signals are synchronized. When the XCLK and DLLCLK signals are synchronized, phase detector 150 activates the phase lock signal PHLOCK. When activated, the PHLOCK signal causes controller 124 to stop adjusting the amount of delay to lock the DLL. The DLL stays in the lock position unless phase detector 150 detects a change in phase between the XCLK and DLLCLK signals. When this occurs, DLL 100 repeats the adjusting and comparison process to ensure that the XCLK and DLLCLK signals remain synchronized.

In the embodiment represented by Figure 1, when the COMMAND signal is not activated while the XCLK and DLLCLK signals are synchronized, the DLLCLK signal is the DLLCK1 signal. This is a normal condition DLL 100, that is, the DLL is locked while the DLLCLK signal is selected from the DLLCK1 signal. However, when the COMMAND signal is activated HIGH due to a change in operating condition of the DLL such as a change in the supply voltage, MUX 130 deselects the DLLCK1 signal and selects the DLLCK0 signal as the DLLCLK signal. In other words, when the COMMAND signal is activated, the DLLCK1 signal is replaced by the DLLCK0 signal as the DLLCLK signal. This replacement occurs before phase detector 150 detects the change by an operating condition of the DLL. Referring again to Figure 2, when the PHDETECT signal is activated at time T2, the DLLCK0 signal has already replaced the

DLLCK1 signal at time T1. This means that MUX 130 selects the DLLCK0 signal in response to the COMMAND signal not in response to phase detector 150. In other words, the DLL switches from the DLLCK1 signal to the DLLCK0 signal before the external and internal clock signals are detected by phase detector 150 as out of

5 synchronism due to the change in the operating condition. Since the COMMAND signal introduces or reduces some delay, and since the DLLCK0 and DLLCK1 signals are provided by two different outputs and having different delays, when the DLLCK1 is replaced by the DLLCK0 as the DLLCLK signal, the DLLCLK and XCLK signals will be out of phase or will not be synchronized. Phase detector 150 detects the phase

10 difference between the XCLK and DLLCLK signals and activates the SL or SR and the PHDETECT signal to begin the adjusting process.

At certain time, the COMMAND signal is deactivated (LOW) which allows the COMMAND_SET signal to be forced LOW when the PHDETECT or PHLOCK signal is asserted (HIGH). When the COMMAND_SET signal is LOW, it causes MUX 130 to

15 reselect the DLLCK1 as the DLLCLK signal. The adjusting process continues until XCLK and DLLCLK signals are synchronized.

In the above description, the switching between the DLLCK0 and DLLCK1 signals is equivalent to switching or changing the amount of delay applied to the XCLK signal. The DLLCK0 and DLLCK1 signals are delayed versions of the XCLK signal.

20 However, the amount of delay applied to the XCLK signal to generate the DLLCK1 signal is not the same as the amount of delay applied to the XCLK signal to generate the DLLCK0 signal. In Figure 1, the number of delay stages being used to apply a delay to the XCLK signal to generate the DLLCK1 signal is greater than the number of delay stages being used to apply the delay to the XCLK signal to generate the DLLCK0

25 signal. Therefore, the amount of delay applied to the XCLK signal to generate the DLLCK1 signal is more than the amount of delay applied to the XCLK signal to generate the DLLCK0 signal.

In summary, when the COMMAND signal is not activated, the first amount of delay is equal to the amount of delay applied to the XCLK signal to generate the

30 DLLCK1 signal (because the DLLCLK signal is selected from the DLLCK1 signal at

this time). When the COMMAND signal is activated, the DLLCK1 signal is replaced by the DLLCK0 as the DLLCLK signal. Because the DLLCLK0 is selected at this time, the amount of delay applied to the XCLK signal at this time is not the same as the first amount of delay. The amount of delay now is equal to a second amount of delay, which is the amount of delay applied to generate the DLLCK0 signal. When the COMMAND_SET signal is deactivated, the DLLCLK signal is again selected from the DLLCK1 signal. Thus, the second amount of delay is changed back to the first amount of delay.

Figure 3 is a block diagram of a delay line 312 connected to a controller 324 according to one embodiment of the invention. Delay line 312 and controller 324 represent delay line 112 and controller 124 of Figure 1. Controller 324 includes a shift register 305 connected to a register control circuitry 320. Shift register 305 includes a plurality of shift register cells 310-0 to 310-N. Register cells 310 0-N include a plurality of outputs 316-0 to 316-N, which are shown as control taps 119-0 to 119-N in Figure 1. The number of shift register cells 310 0-N is equal to the number of delay stages 113 0-N of Figure 1. Control circuitry 320 connects to lines 342, 343 and 344 to receive the SL, SR and PHLOCK signals, which are represented by the SL, SR and PHLOCK signals on lines 142, 143 and 144 of Figure 1.

In operation, control circuitry 320 activates a predetermined number output lines 316 0-N. Based on the activated output lines, delay line 312 selects a predetermined number of delay stages to apply an initial amount of delay to the XCLK signal. In a subsequent operation, register control circuitry 320 receives the SL or SR signals provided on lines 342 or 343. Based on the SL or SR signal, control circuitry 320 activates different output lines 316 0-N. Based on the selected output lines, delay line 312 selects a different number of delay cells 313 0-N that the XCLK signal passes through. This changes the amount of delay applied to the XCLK signal accordingly. Control circuitry 320 and delay line 312 repeat the process until the XCLK and DLLCLK signals are synchronized. The DLLCLK signal is selected from either the DLLCK0 or the DLLCK1 depending on whether the COMMAND signal is activated.

Figure 4A is a block diagram of a delay line 400A according to another embodiment of the invention. Delay line 400A includes a plurality of delay stages 413 0-N, which are similar to delay stages 113 0-N of delay line 112 of Figure 1. Delay line 400A also provides the DLLCK0 and DLLCK1 signals from the outputs of the delay lines that are located in the same positions as that of delay line 112. However, the orders of the DLLCK0 and DLLCK1 signals of delay line 400A are switched at MUX 130. That is, the DLLCK0 signal is provided to input B of MUX 130 and the DLLCK1 signal is provided to input A of MUX 130. In this arrangement, the DLLCLK signal is normally selected from the DLLCK0, whereas in Figure 1, the DLLCLK is normally selected from DLLCK1. Therefore, when the COMMAND signal is activated, the amount of delay is increased. In contrast, in Figure 1, when the COMMAND signal is activated, the amount of delay is decreased.

Figure 4B is a block diagram of a delay line 400B according to another embodiment of the invention. Delay line 400B is similar to delay line 400A shown in Figure 4B except that the delay difference between the DLLCK0 and DLLCK1 signals of delay line 400B is equal to the delay of more than one delay stages. In delay line 400B, the DLLCK0 is provided by the output of a delay line 413 (N-X), where X is equal to or greater than two. For example, if X is 2, then the DLLCK0 signal is provided by the output of delay stage 413 (N-2), which is the delay stage located 2 delay stage positions preceding the last delay stage 413-N. Since the DLLCK1 signal is provided by the last delay stage 413-N, the difference in delay between the DLLCK1 and DLLCK0 signals is equal to the delay of two delay stages. Similar to delay line 400A, the DLLCK0 of delay line 400B is normally selected as the DLLCLK signal when the COMMAND signal is not activated. Therefore, when the DLLCK1 signal is selected as the DLLCLK signal while the COMMAND signal is activated, the amount of delay is increased by a delay equal to the delay of two delay stages. Following the same pattern, if X is greater than two, then the difference in delay between the DLLCK1 and DLLCK0 signals is equal to the delay of one delay stage multiplied by X.

Figure 4C is a block diagram of a delay line 400C according to another embodiment of the invention. Delay line 400C is similar to delay line 400B shown in

Figure 4B except that the orders of the DLLCK0 and DLLCK1 signals of delay line 400C are switched at MUX 130. That is, the DLLCK0 signal is provided to input A of MUX 130 and the DLLCK1 signal is provided to input B of MUX 130. This arrangement is similar to that of delay line 112 shown in Figure 1 in which the

- 5 DLLCLK signal is normally selected from the DLLCK1 when the COMMAND signal is not activated. Therefore, when the DLLCK0 signal is selected as the DLLCLK signal while the COMMAND signal is activated, the amount of delay is decreased by a delay equal to the delay of two delay stages if $X = 2$. Similar to delay line 400B, if X is greater than two, then the difference in delay between the DLLCK1 and DLLCK0
10 signals is equal to the delay of one delay stage multiplied by X .

- Figures 4A-C show the various embodiments of a delay line connected to a selector. However, other embodiments similar to the ones shown in Figures A-C and also be constructed to achieve the same purpose. For example, in the embodiment represented by Figure 4D, a delay line 400D has the DLLCK0 and DLLCK1 signals
15 derived from the same output of a delay cell such as delay cell 413-N. In this embodiment, however, the DLLCK0 signal is passed through a delay element, such as delay element 444, before it is inputted to MUX 130. The delay time of delay element 444 can be preset such that a difference in delay between the DLLCK0 and DLLCK1 signals is equal to a predetermined delay.

- 20 Figure 4E shows another embodiment of a delay line 400E connected to a selector. In the embodiment represented by Figure 5, MUX 130 includes multiple inputs A, B and C, and multiple select lines 450, 451 and 452. Inputs A, B and C connect to different outputs of different delay cells such as outputs of delay cells 413 (N-2), 423 (N-1) and 413-N. In this embodiment, multiple delayed signals such as the
25 DLLCK0, DLLCK1, and DLLCK2 signals can be selected based on the activation of different signals S0, S1 and S2 on lines 450, 451 and 452.

- Figure 5 is schematic diagram of a command react circuit 500 according to one embodiment of the invention. Command react circuit 500 is similar to command react circuit 140 of Figure 1. Command react circuit 500 includes a first input connected to
30 line 544 to receive a phase detect signal PHDETECT, a second input connected to line

546 to receive a command signal COMMAND, and an output connected to line 522 to provide a select signal COMMAND_SET. The signals at inputs 544 and 546, and output 522 are similar to the signals with the same names shown in Figure 1.

The output of command react circuit 500 includes a memory unit 510, which has

5 a first latch node connected to line 522 of a memory unit 510. Memory unit 510 also includes a pair of inverters 512 and 514. A second latch node of memory unit 510 connects to node 515. Latch node 515 connects to a supply voltage Vcc at node 511 through two series-connected p-channel transistors 532 and 534. A NOR gate 543 has inputs connected to lines 544 and 545 to receive the PHDETECT and PHLOCK signals.

10 An output of NOR gate 543 connects to a gate of transistor 532. A gate of transistor 534 connects to line 546 to receive the COMMAND signal. Latch node 515 also connects to ground at node 513 through an n-channel transistor 550. A gate of transistor 550 connects to line 546 to receive the COMMAND signal.

Command react circuit 500 provides the select signal COMMAND_SET in

15 response to the PHDETECT, PHLOCK and COMMAND signals. In Figure 1, based on the COMMAND_SET signal, MUX 130 selects either the DLLCK0 or DLLCK1 signal as the DLLCLK signal. In Figure 5, the COMMAND signal is not activated (inactive) when it is LOW; the COMMAND signal is activated (active) when it is HIGH.

When the COMMAND signal is LOW, it turns off transistor 550 and turns on

20 transistor 534. When transistor 550 is off, latch node 515 is disconnected from ground. When the PHDETECT or PHLOCK signal is HIGH, it forces the output of NOR gate 543 LOW which turns on transistor 532. When both transistors 532 and 534 are on, latch node 515 is connected to Vcc (or HIGH) forcing latch node on line 522 LOW. This means the COMMAND_SET signal is LOW. In Figure 1, in response to

25 the COMMAND_SET signal being LOW, MUX 130 selects the DLLCK1 signal as the DLLCLK signal.

When the COMMAND signal is HIGH, it turns on transistor 550 and turns off transistor 534. When transistor 534 is off the path connecting Vcc to latch node 515 is cut off regardless of the states of the PHDETECT and PHLOCK signals. When

30 transistor 550 is on, latch node 515 is connected to ground forcing latch node at line

522 HIGH. This means the COMMAND_SET signal is HIGH. In Figure 1, in response to the COMMAND_SET signal being HIGH, MUX 130 selects the DLLCK0 signal as the DLLCLK signal. When the COMMAND signal goes LOW, the COMMAND_SET signal stays HIGH until the PHDETECT or PHLOCK signal goes HIGH.

In summary, when the COMMAND signal is LOW and the PHDETECT or PHLOCK signal has been HIGH at some point after the COMMAND signal last went HIGH, the DLLCK1 signal is selected as the DLLCLK signal, and when the COMMAND signal is HIGH, the DLLCK0 signal is selected as the DLLCLK signal until the COMMAND set signal is LOW and the either the PHDETECT or PHLOCK signal goes HIGH.

Figure 6 is a block diagram of a memory device 600 according to the invention. Memory device 600 includes a plurality of memory cells 602 arranged in rows and columns. Row decode circuit 604 and column decode circuit 606 access the rows and columns in response to an address, provided on a plurality of address lines or bus 608. Data is transferred to and from memory device 600 through data lines or bus 610. A command decode circuit 616 controls data communication to and from memory device 600, via input circuit 620 and output circuit 622 in response to input signals on control lines 614. Command decode circuit 616 receives the input signals on lines 614 to activates a plurality of command signals COMMAND-0 to COMMAND-X on lines 644-0 to 644-X. The activations of the command signals determine the modes of operation of memory device 600 such as active, refresh, read, and write. The input signals on lines 614 include, but are not limited to, External Clock (XCLK), Row Access Strobe (RAS*), Column Access Strobe (CAS*), and Write Enable (WE*).

In addition, memory device 600 also includes a DLL 601. DLL 601 is similar to DLL 100 shown in Figure 1. During a memory mode such as an ACTIVE mode or a REFRESH mode, DLL 601 replaces an internal generated clock signal, such as the DLLCLK signal, with a signal such as the DLLCK0 or DLLCK1 signal, that has a different amount of delay in relation to the external clock signal XCLK.

In operation, a certain combination of the input signals, such as the RAS, CAS and WE signals, causes command decode circuit 616 to activate one of the COMMAND-0 to COMMAND-X signals. One of these COMMAND-0 to COMMAND-X signals is represented by the COMMAND signal shown on line 146 of Figure 1. The activated command signal is provided to DLL 601 such that the DLL operates in a similar fashion of DLL 100 of Figure 1.

Memory device 600 of Figure 6 can be a dynamic random access memory (DRAM) or other types of memory circuits such as SRAM (Static Random Access Memory) or Flash memories. Furthermore, the DRAM could be a synchronous DRAM commonly referred to as SGRAM (Synchronous Graphics Random Access Memory), SDRAM (Synchronous Dynamic Random Access Memory), SDRAM II, or DDR SDRAM (Double Data Rate SDRAM), as well as Synchlink or Rambus DRAMs. Those of ordinary skill in the art will readily recognize that memory device 600 of Figure 6 is simplified to illustrate one embodiment of a memory device of the present invention and is not intended to be a detailed description of all of the features of a memory device.

Figure 7 is a block diagram showing in more detail a portion of memory device 700. Memory device 700 is similar to memory device 600 of Figure 6. In Figure 7, command decode circuit 716 includes inputs connected to lines 714 to receive the XCLK, RAS, CAS and WE signals, and a plurality of outputs connected to line 744-0 to 744-M, which provide a plurality of command signals COMMAND-0 to COMMAND-X. The COMMAND-X signal is provided to a row line driver or word line driver 720. The COMMAND-X signal is also provided to DLL 701. The COMMAND-X signal is represented in Figure 1 as the COMMAND signal. Driver 720 connects to the gates of a plurality of access transistors 730 via line 722 to provide a control signal ACT. Transistors 730 and memory cells 740 are arranged in rows 735-0 to 735-N and in columns 745-0 to 745-M.

In operation, different combinations of the RAS, CAS and WE signals activate different command signals on lines 744-0 to 744-M. In one combination of the RAS, CAS and WE signals, command decode circuit 716 activates the COMMAND-X signal

on line 744-M to put the memory device to a certain mode. For example, when RAS, CAS and WE have a combination of LOW, HIGH, and HIGH, memory device is put in an ACTIVE mode. In this mode, a row of memory cells such as row 735-0 is activated or "opened" in preparation for a subsequent READ, WRITE or other modes. In another
 5 embodiment, a column of memory cells such as column 745-0 is activated.

When command decode circuit 716 receives the right combination for a memory mode, such as the memory ACTIVE mode, the COMMAND-X signal is activated. Driver 720, in response to the activated COMMAND-X signal, activates the ACT signal on line 722. The activated ACT signal turns on transistors 730 to activate or open the
 10 row of memory cells 740. While driver 720 responds to the COMMAND-X signal, DLL 701 also reacts to the COMMAND-X signal in a manner similar to the manner in which DLL 100 reacts to the COMMAND signal as described in Figures 1-3.

In the embodiment represented by Figure 7, to concentrate on the invention, only one example is shown to illustrate how a command signal such as the COMMAND-X
 15 signal is used during one particular mode of memory device 700 such as the ACTIVE mode. However, command signals such as the COMMAND-0, COMMAND-1 and others can also be used in connection with DLL 701 or with other DLLs similar to DLL 701 of memory device 700 in other modes. Although different command signal represents different modes of the memory device, all command signals affect the
 20 operation of a DLL such as DLL 100 or 701 in the same manner. That is, the command signal causes the DLL to change the amount of delay applied to the XCLK signal by either increasing or decreasing the amount of the delay when the command signal is activated.

For example, the COMMAND-0 or COMMAND-1 signal can be a READ or
 25 REFRESH command signal. The READ command signal is activated during a read mode to read data stored in memory cells such as memory cells 740. The REFRESH command is activated during a refresh mode to refresh the data in memory cells such as memory cells 740 to ensure that the memory cells retain their valid data values. The activation of the read or refresh mode can vary the supply voltage of the memory device
 30 which also affects the operation of the DLL in that it speeds up or slows down

DLLCLK. To compensate for the variation in the supply voltage, during the read or refresh mode, the COMMAND-0 (READ) or COMMAND-1 (REFRESH) signal can be provided as the COMMAND signal shown in Figure 1 so that the DLL can select the DLLCLK signal from signals such as the DLLCK0 and the DLLCK1 signals to ensure
5 that the variation in the voltage supply is properly compensated during the read or refresh mode.

Figure 8 shows a system 800 according to the invention. System 800 includes a processor 802 and a memory device 804. Memory device 804 represents memory device 600 or 700 of the invention. Processor 802 can be a microprocessor, digital
10 signal processor, embedded processor, microcontroller, or the like. Processor 802 and memory device 804 communicate using address signals on lines 808, control signals on lines 810, and data signals on lines 806.

Memory device 804 includes a DLL 801. DLL 801 is similar to DLL 100 described in the figures above (include all embodiments). According to the invention,
15 during a memory operation, processor 802 provides certain input signals to memory device 804 via lines 810. The input signals are similar to the RAS, CAS and WE signals shown in Figures 6 and 7. Thus, certain combination of the input signals causes memory device 804 to operate in a certain mode such as the ACTIVE mode. The operation of memory device 804 including the operation of DLL 801 is describe in
20 connection with Figures 1-7.

Conclusion

A novel delay locked loop (DLL) has been described. The DLL applies an amount of delay to an external clock signal to generate a first and a second delayed signal. One of the delayed signals is selected as an internal clock signal. The first and
5 second delayed signals have different delays in relation to the external clock signal. If a change in operating condition of the DLL occurs, such as a change in the supply voltage during an operational mode of the memory device such as an ACTIVE, a READ or a REFRESH mode, the DLL immediately selects another delayed signal as a new internal clock signal to compensate for the change in operating condition before a phase detector
10 of the DLL detects the change.

Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiment shown. This application is intended to cover any adaptations or variations of the present
15 invention. Therefore, it is intended that this invention be limited only by the claims and the equivalents thereof.

09903227-071101